UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

2

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

2a

AD-A264 961

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4

5 MONITORING ORGANIZATION REPORT NUMBER(S)
#67

6a NAME OF PERFORMING ORGANIZATION	6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION
Institute for Brain and Neural Systems		Personnel and Training Research Programs Office of Naval Research (Code 1142PT)

6c. ADDRESS (City, State, and ZIP Code)	7b ADDRESS (City, State, and ZIP Code)
Brown University Providence, Rhode Island 02912	800 North Quincy Street Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
		N00014-91-J-1316

8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

DTIC ELECTE MAY 26 1993 S E D

11 TITLE (Include Security Classification)
Unsupervised Splitting Rules for Neural Tree Classifiers.

12 PERSONAL AUTHOR(S)
Michael P. Perone and Nathan Intrator

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Report	FROM _____ TO _____	May 17, 1993	Six

16. SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	CART, Unsupervised Feature Extraction, Neural Trees
05	08		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper presents two unsupervised neural network splitting rules for use with CART-like neural tree algorithms in high dimensional data space. These splitting rules use an adaptive variance estimate to avoid some possible local minima which arise in unsupervised methods. We explain when the unsupervised splitting rules outperform supervised neural network splitting rules and when the unsupervised splitting rules outperform the standard node impurity splitting rules of CART. Using these unsupervised splitting rules lead to a nonparametric classifier for high dimensional space that extracts local features in an optimized way.

93 5 25 058

93-11614

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☑ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Dr. Joel Davis	(703) 696-4744	

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Unsupervised Splitting Rules for Neural Tree Classifiers*

Michael P. Perrone and **Nathan Intrator**
Center for Neural Science
Brown University
Providence, RI 02912
E-mail: mpp@cns.brown.edu and nin@cns.brown.edu

June 6, 1992

Abstract

This paper presents two unsupervised neural network splitting rules for use with CART-like neural tree algorithms in high dimensional data space. These splitting rules use an adaptive variance estimate to avoid some possible local minima which arise in unsupervised methods. We explain when the unsupervised splitting rules outperform supervised neural network splitting rules and when the unsupervised splitting rules outperform the standard node impurity splitting rules of CART. Using these unsupervised splitting rules leads to a nonparametric classifier for high dimensional space that extracts local features in an optimized way.

1 Introduction

Due to the *curse of dimensionality* (Bellman, 1961) it is desirable to extract features from a high dimensional data space before attempting a classification. This may be done in those cases where the important structure is assumed to lie in a low dimensional subspace of the original data. A general method for feature extraction is Projection Pursuit, and its unsupervised version - Exploratory Projection Pursuit (Friedman and Tukey, 1974; Friedman, 1987).

One of the advantages of EPP is the use of locally smooth objective functions in the search for interesting features. Such functions are not related to the class labels, and have the potential of avoiding the curse of dimensionality (Huber, 1985). The method has an underlying assumption of homogeneity of the input space. Intuitively this means that a useful feature can only be found based on all of the input patterns. This poses a disadvantage due to the fact that the labels are not used through the search for good projections, and therefore, it is possible to ignore features that may only be important for classifying a small portion of the input data but are less interesting when considering the data as a whole. This observation is one of the motivations of recursive partitioning methods, including tree structured algorithms.

2 CART-based Neural Trees

CART addresses high dimensional space problems by partitioning the space and replacing complex classifiers (or regressors) designed for the whole input space, by a set of simpler modules working on smaller subregions of the space. There have been some recent attempts for recursive partitioning classification [see for example (Jacobs et al., 1991; Sankar and Mammone, 1991; Intrator, 1991; Perrone, 1991)].

CART is not directly applicable to classification problems in very high dimensional spaces, such as gray level pixel images, since splitting based on a single dimension (single pixel in this case) is unlikely to increase

*This work was supported in part by the National Science Foundation, the Office of Naval Research, and the Army Research Office.

the homogeneity of sub regions in the space. At each node of the tree, additional features are sought before the split is constructed, using only that portion of the input space that arrives to this node, and these new features are added to the features extracted so far, to construct an optimal split at that node. This leads to a combination of feature extraction and recursive partitioning that has the potential to be much more powerful than each of the methods by itself. Moreover, this method is still consistent with the monotonicity requirement of the cost at each split (Breiman et al., 1984), and therefore allows the use of the powerful pruning mechanism of CART.

Thus, by combining these techniques, one can develop a hybrid neural tree algorithm. The construction of the hybrid tree then proceeds the same as in the CART method (Breiman et al., 1984) with the exception that every node can perform additional feature extraction based on the high dimensional input patterns that arrive at that node, and based on the features extracted so far. The construction of a nested sequence of trees, the pruning based on cost, complexity cross-validation and the final tree selection can all be done exactly in the same way as in CART.

3 Comparison with CART Splitting Rules

It is important to see that the splitting rules presented in this paper have the desired feature extracting behavior. To see this behavior, we compare them to some splitting rules commonly used with CART (Breiman et al., 1984). Let $p(i|t)$ be the probability of class i at a tree node t and let p_L and p_R be the percentages of the data set on the 'left' and 'right' sides of a given partition. Some of the common CART splitting rules can be written in terms of these quantities as follows.

$$I_{\text{Entropy}}(t) = -\sum_{s=L,R} p_s \sum_i p(i|t_s) \ln p(i|t_s),$$

$$I_{\text{Gini}}(t) = \sum_{s=L,R} p_s \sum_{i \neq j} p(j|t_s) p(i|t_s),$$

$$I_{\text{Twoing}}(t) = \frac{p_L p_R}{4} \Big[\sum_j |p(j|t_L) - p(j|t_R)|\Big]^2,$$

$$I_{\text{Delta}}(t) = 2 p_L p_R \Big[\sum_j |p(j|t_L) - p(j|t_R)|\Big].$$

The optimal split is then found by minimizing one of these measures.

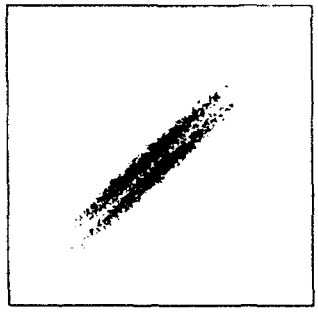

Figure 1: Randomly generated gaussian clusters. Each cluster corresponds to one class

For the double gaussian distribution shown in figure 1, the CART measures are shown in figure 2. In this example, each cluster represents a different class. It is easy to see that these measures find the best split. In figure 5, the unsupervised splitting rules are shown for the same double gaussian distribution. It should be noted that the structure found by the CART measures depend completely on the label; whereas the neural splitting costs don't have this restriction.

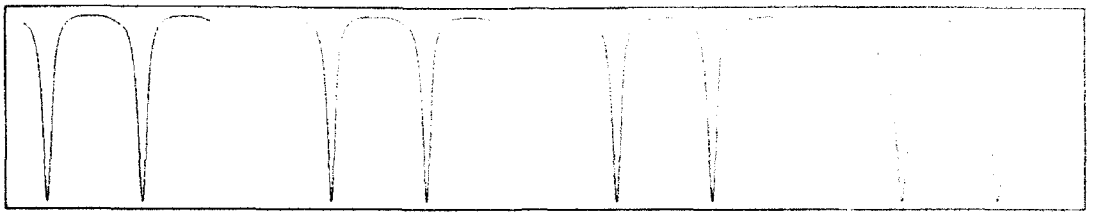

Figure 2: From left to right: the Gini, Entropy, Twoing and Delta measures. These measures correspond only to splits which pass through the center of the data shown in figure 1. The rotation angle of the split is plotted on the x-axis from 0 to 2π. Each measure is minimized at $\frac{\pi}{4}$ and $\frac{5\pi}{4}$.

4 Supervised vs Unsupervised Features

An example where this splitting rule along with feature extraction may be useful is given in figure 3. It shows a subregion in space in which two classes are strongly mixed. A supervised splitting algorithm will split according to hyperplane 1 whereas the above unsupervised splitting rule will prefer to split according to hyperplane 2. This is because split 1 increases the purity of each node more than split 2 although split 1 does not focus on the confusion region between class A and B. It is conceivable that if the confused region is transferred in full to a node, and then an attempt to extract more informative features only from this region is made, the new representation will have a better chance to reduce the confusion between the classes in this subregion.

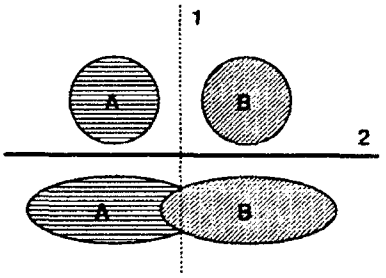

Figure 3: The ability of an unsupervised splitting rule to reduce confusion.

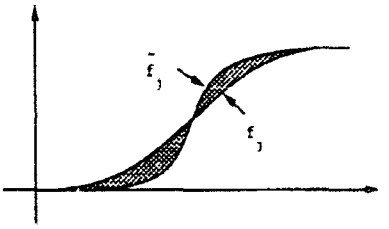

Figure 4: The minimization of the pseudo-supervised MSE, is equivalent to minimizing the shaded area in the picture.

5 Generalized Pseudo-Supervised Splitting Rule

In (Intrator, 1991), a pseudo-supervised variant of backpropagation was presented for finding optimal splits for a neural tree classifier. A related unsupervised technique is presented in (Bridle and Cox, 1991). In this section, we extend Intrator's results to a more robust splitting rule. For simplicity, we shall follow the notations presented in (Rumelhart et al., 1986).

Let o_{pj} be the output of the j'th splitting rule function for input pattern p. f_j is a sigmoidal activation function defined by $f_j(t) = [1 + \exp(-t)]^{-1}$, so that $o_{pj} = f_j(net_{pj})$, where $net_{pj} = \frac{1}{\sigma_j} \sum_i w_{ji} o_{pi}$ and ${\sigma_j}^2 = \mathrm{var}_p(\sum_i w_{ji} o_{pi})$. Let the target for output j be also defined in terms of the network activity, $t_{pj} = \tilde{f}_j(net_{pj})$, where $\tilde{f}_j$ is a sigmoidal function with a gain constant $\lambda > 1$, $\tilde{f}_j(t) = [1 + \exp(-\lambda t)]^{-1}$. The network is trained to minimize the empirical MSE $\sum_p (t_p - o_p)^2$. In order to avoid trivial splits it is possible to add penalty of

the form

$$\kappa\big[1-\big(\frac{1}{n}\sum_p o_p\big)\big(\frac{1}{n}\sum_p(1-o_p)\big)\big],$$

for some small constant κ, however, simulations show that the trivial split does not usually happen especially when there are several neurons in the hidden and output layer.

The difference between t_{pj} and o_{pj} is shown in figure 4. This target function approximates a characteristic function, an approximation which will improve when $\lambda \rightarrow \infty$. In practice, there is no need to have λ be greater than 5. The calculation of the gradient with respect to the weight w_{ij} follows in the same way as in (Rumelhart et al., 1986), when taking into account the fact that the target depends on the network output as well. For an output layer unit j we have

$$-\frac{\partial E_p}{\partial w_{ji}} = -\big[\frac{\partial E_p}{\partial o_{pj}}\frac{\partial o_{pj}}{\partial net_{pj}} + \frac{\partial E_p}{\partial t_{pj}}\frac{\partial t_{pj}}{\partial net_{pj}}\big]\frac{\partial net_{pj}}{\partial w_{ji}},$$

and it follows that for

$$\delta_{pj} \equiv \frac{1}{\sigma_j}(t_{pj}-o_{pj})\big[o_{pj}(1-o_{pj}) - \lambda t_{pj}(1-t_{pj})\big],$$

we get

$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} o_{pi}.$$

The calculation of the gradient with respect to a hidden unit weight is exactly as in (Rumelhart et al., 1986), and will not be repeated here.

An intuitive explanation to this target definition is similar to the reasoning behind hard and soft competition approaches (Hinton and Nowlan, 1990). If a hard target (0 or 1) is imposed, then whenever the output is close to .5 which means that the input is close to the boundary, the error signal would be large. However if the input is close to the boundary, it is likely to be on the wrong side of the boundary, which will then lead to a large wrong correction signal. Using the soft target which takes into account the confidence in the output solves this problem, since the target is also close to 0.5. Another explanation is obtained by observing that the target is also dependent on the synaptic weights, and therefore the gradient of the synaptic weights with respect to the output should be taken into account as well. This requires the use of a soft target.

The construction of a binary splitting rule based on the above criterion is done by letting the PS network converge (or stop training based on another criterion) and then assign the patterns for which the output of the network is greater than .5 to t_R. In the case of a multi-split, assign to set j the patterns for which the output of unit j in the network is greater than .5.

6 Gaussian Splitting Rule

In this section we present a variation of the splitting rule described above. For this splitting rule, we define a splitting cost by replacing the sigmoidal activation functions, $o_{pj} = f_j(net_{pj})$ of the final layer of a backprop network with gaussians, $o_{pj} = g_j(net_{pj})$ where $g_j(t) \equiv exp(-t^2/2{\sigma_j}^2)$, $net_{pj} = \sum_i w_{ji}o_{pi}$ and ${\sigma_j}^2 = \mathrm{var}_p(net_{pj})$. In the case, the splitting cost, E, is defined as

$$E \equiv \sum_{p,j} o_{pj}.$$

So the delta rule for a pattern p is given by

$$-\frac{\partial E_p}{\partial w_{ji}} = -\frac{\partial E_p}{\partial o_{pj}}\frac{\partial o_{pj}}{\partial net_{pj}}\frac{\partial net_{pj}}{\partial w_{ji}},$$

and it follows that for

$$\delta_{pj} \equiv \frac{1}{{\sigma_j}^2} o_{pj}\, net_{pj},$$

we get

$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} o_{pi}.$$

Again we have that the hidden unit gradients follow exactly as in (Rumelhart et al., 1986) and can be implemented with a minor modification to a backpropagation algorithm.

For the splitting rule outlined above, the cost, E, is minimized by pushing the data points to either side of the central gaussian. It is in this way that the splitting rule extracts clusters from high dimensional spaces. As in the previous section, we can add the same penalty to avoid trivial splits.

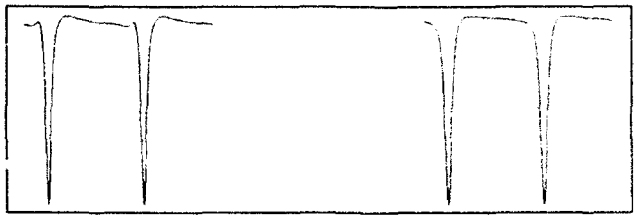

Figure 5: Pseudo-supervised and unsupervised splitting measures. These measures correspond only to splits which pass through the center of the data shown in figure 1. The rotation angle of the split is plotted on the x-axis from 0 to 2π. Each measure is minimized at $\frac{\pi}{4}$ and $\frac{5\pi}{4}$.

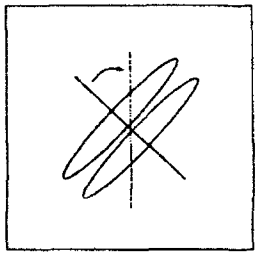

Figure 6: The split orthogonal to the optimal split is a local minimum because a rotation in either direction will increase the cost.

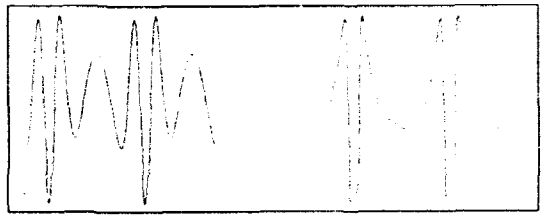

Figure 7: Local minima for the pseudo-supervised and unsupervised gaussian splitting rules dominate the parameter space with the exception of small *squint angles* which have maintained global minima.

7 Avoiding Local Minima

Unsupervised splitting methods suffer from spurious local minima that do not exist for supervised algorithms (Figure 6). This problem is amplified dramatically in high dimensional spaces where the *squint angle* (i.e. the solid angle on a unit hypersphere for which a projection reveals the clustering structure) becomes extremely narrow. (See (Huber, 1985) for a detailed discussion of the statistical problems involved.) In these cases, nearly every minima is a local minima.

However, the unsupervised splitting rules presented in this paper avoid this problem by using an empirical variance to remove the local minima shown The unsupervised splitting rules without the variance modification are shown in figure 7. Due to the narrow squint angle, the local minima occupy most of the parameter space.

8 Discussion

A method of recursive partitioning for high dimensional input spaces was introduced. This was done by combining the benefits from exploratory projection pursuit with those from the CART method. New exploratory splitting rules were presented, and argued to have the potential to be less biased to the training data. The splitting rules, can have a boundary that contains an arbitrary predefined number of hyperplanes by defining the number of hidden units in the feedforward network, and is easily extended into multiple splits. In addition, the splitting rules avoid some of the local minima of other unsupervised splitting rules and they are not plagues by the problems with exhaustive searches in high dimensional spaces.

Combining all the above ingredients together, results in a computationally practical method for non-parametric classification in very high dimensional spaces, that is less sensitive to the curse of dimensionality due to the feature extraction, and is less biased to the training data, due to the sophisticated tree construction of the CART method.

References

Bellman, R. E. (1961). *Adaptive Control Processes*. Princeton University Press, Princeton, NJ.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. The Wadsworth Statistics/Probability Series, Belmont, CA.

Bridle, J. S. and Cox, S. J. (1991). RecNorm: simultaneous normalization and classification applied to speech recognition. In *Advances in Neural Information Processing Systems 3*.

Friedman, J. H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266.

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C(23):881–889.

Hinton, G. E. and Nowlan, S. J. (1990). The bootstrap widrow-hoff rule as a cluster-formation algorithm. *Neural Computation*, 2(3):355–362.

Huber, P. J. (1985). Projection pursuit. (with discussion). *The Annals of Statistics*, 13:435–475.

Intrator, N. (1991). Localized exploratory projection pursuit. In Wegman, E., editor, *Computer Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 237–240. Amer. Statist. Assoc., Washington, DC.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Perrone, M. P. (1991). A novel recursive partitioning criterion. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, Cambridge, MA.

Sankar, A. S. and Mammone, R. J. (1991). Neural tree networks. In Mammone, R. J. and Zeevi, Y., editors, *Neural Networks: Theory and Apllications*. Academic Press, New York.